

Third Day

التكرار Loop

تكرار العملية باستخدام for

قد نحتاج في بعض الحالات إلى تكرار مجموعة من التعليمات البرمجية عدد من المرات، 10، 20 ، أو حتى 100 مرة أو أكثر، فبدلاً من تكرار كتابة نفس التعليمات البرمجية يمكننا استخدام for.

```
int len = 5;

for(int i = 0; i < len; i++){

    System.out.println(i);

}
```

المخرجات

```
0
1
2
3
4
```

تكرار العملية باستخدام while

يمكننا تكرار تنفيذ أمر برمجي وفقاً لشرط معين باستخدام while على النحو التالي:

- كتابة المتغير الذي سيتم قياس تحقق الشرط على قيمته
- استخدام while لكتابة الشرط الذي سيتم تكرار تنفيذ الأمر حال تحققه وهو أن تكون قيمة المتغير number أكبر من

أوتساوي 1

- الأوامر التي سيتم تكرار تنفيذها حال تحقق الشرط وهي كالتالي:

a. طباعة قيمة المتغير `number`.

طرح 1 من المتغير `number`

```
int number = 5;

while(number >= 1){

    System.out.println(number);

    number--;

}
```

المخرجات

```
5
4
3
2
1
```

تكرار العملية باستخدام do-while

يمكننا تكرار تنفيذ أمر برمجي وفقاً لشرط معين باستخدام `do-while` على النحو التالي:

- تختلف `do-while` عن `while` بحيث انها سوف تنفذ الأمر الموجود بداخلها مرة واحدة بغض النظر عن الشرط

في المثال التالي سوف يكون الشرط ان `number >= 1` و لكن قيمة `number` تساوي -1 حيث انه لن يتم تنفيذ الشرط

```
int number = -1;

do{

    System.out.println(number);

    number--;

}while(number >= 1);
```

-1

إيقاف التكرار واستخدام break

تقوم break بإنهاء عملية التكرار بشكل كامل . يوضح المثال التالي كيفية الخروج من for عندما نحصل على الرقم الزوجي الأول:

```
for(var i = 1; i < 10; i++){
    if(i % 2 ==0 ){
        break;
    }
    System.out.println(i);
}
```

تجاوز خطوة من التكرار واستخدام continue

تقوم continue بإيقاف التعليمات البرمجية و الانتقال لتكرار التعليمات البرمجية للعنصر التالي . يوضح المثال التالي كيف يمكننا استخدام عبارة continue لطباعة الأرقام الفردية فقط :

```
for(var i = 1; i < 10; i++){
    if(i % 2 ==0){
        continue;
    }
    System.out.println(i);
}
```